

Listing of Claims:

1. (currently amended) A method for providing parallel execution of computing tasks in a heterogeneous computing environment having a plurality of computing hosts, comprising:

partitioning a computing task into small tasks;

assigning each of the small tasks to mobile agents stored at a central server;

monitoring the computing hosts for detecting an indication that migration needs to be initiated;

determining available computing hosts in the heterogeneous computing environment;

automatically transferring said mobile agents to determined available computing hosts responsive to a detected indication that migration needs to be initiated;

executing said mobile agents at the available computing hosts using execution code ~~provided to said mobile agents~~ from a the central server in response to a request from the available computing hosts; and

maintaining, at a second computing host at which none of said mobile agents are executing, stack trace and state information about each of said mobile agents to allow one or more of said mobile agents to be reconstructed at an alternate computing host using said stack trace and state information.

2. (previously amended) The method of claim 1, wherein said indication that migration needs to be initiated comprises an indication of network latency.

3. (currently amended) The method of claim 1, further comprising transferring the execution code from the central server to virtual machines at the available computing hosts.

4. (original) The method of claim 1, wherein the step of executing the mobile agents is performed in virtual machines at the computing hosts without modification of the virtual machines.

5. (cancelled)

6. (currently amended) The method of claim 3, wherein the step of transferring the execution code is performed in response to a request by a mobile agent.

7. (currently amended) The method of claim 6, wherein the step of transferring the execution code is performed by a web server.

8. (original) The method of claim 1 further comprising monitoring execution of the mobile agents at the available computing hosts.

9. (previously presented) The method of claim 1 further comprising allowing the mobile agents to collaborate over the heterogeneous computing environment as the mobile agents execute at the available computing hosts.

10. (original) The method of claim 1, wherein the mobile agents are executed in real time.

11. (currently amended) A method for providing parallel computing using mobile agents in a heterogeneous computing environment having a plurality of computing hosts, comprising:

assigning a computing task to one or more mobile agents stored at a central server;

monitoring the computing hosts for detecting an indication that migration needs to be initiated;

transferring the one or more mobile agents to one or more available computing hosts responsive to a detected indication that migration needs to be initiated;

transferring execution code from a the central server to the one or more available computing hosts in response to a request from the available computing hosts;

executing the one or more mobile agents at the one or more available computing hosts using the execution code; and

storing stack trace and real-time state information about the one or more mobile agents at a first alternate computing host on which none of the one or more mobile agents are executing to allow the one or more mobile agents to be reconstructed at a second alternate computing host.

12. (currently amended) The method of claim 11, wherein a said detected indication that migration needs to be initiated is an indication of network latency, said method further comprising, prior to transferring the mobile agents, the step of:

halting transferring of the mobile agents if said detected network latency exceeds a threshold.

13. (previously amended) The method of claim 12 comprising transferring the mobile agents if utilization of computing hosts fails to exceed a predetermined threshold.

14. (currently amended) A method for migrating a software application running in a virtual machine from a primary host to a secondary host comprising:

constructing an application using a plurality of mobile agents stored in a central server;

transferring the plurality of mobile agents to a first computing host;

executing the plurality of mobile agents at said first computing host;

maintaining stack trace and state information about each of the plurality of mobile agents at a second computing host on which none of the plurality of mobile agents are executing;

detecting an indication to migrate the application, wherein said indication comprises one of the group consisting of: network latency, hostile attack, hacking, network failure ~~or~~ and computer hardware failure; ~~and~~

in response to a detected indication, automatically migrating the application in its entirety from said first computing host to a third computing host without modifying the virtual machine at said third computing host by reconstructing each of the plurality of mobile agents at said third computing host using said stack trace and state information[.] ;
and

executing the plurality of mobile agents at the third computing host using execution code provided by the central server in response to a request from the third computing host.

15. (previously amended) The method of claim 1, wherein said step of determining available computing hosts is performed in real time.

16. (previously amended) The method of claim 1, wherein the indication comprises an indication of hostile attack.

17. (previously amended) The method of claim 1, wherein the indication comprises an indication of hacking.

18. (previously amended) The method of claim 1, wherein the indication comprises an indication of network failure.

19. (previously amended) The method of claim 1 wherein the indication comprises an indication of computer hardware failure.

20. (previously amended) The method of claim 14 further comprising resuming execution of the mobile agent at said third computing host at a point where execution was halted.

21. (currently amended) The method of claim 20, wherein said stack trace and state information comprises information about an execution thread of the mobile agent as it existed at said first computing host prior to being transferred to said third computing host.

Claims 22-24 (cancelled)

25. (original) The method of claim 14 further comprising:
continuing monitoring for another indication to migrate the application;
continuing migrating the application to other hosts.

26. (currently amended) A computer system for providing parallel execution of computing tasks in a heterogeneous computing environment comprising at least two computing hosts, said system comprising:

a dispatcher for partitioning the computing task into a plurality of small tasks and dispatching the small tasks;

mobile agents for receiving small tasks from the dispatcher;

means for monitoring execution of the mobile agents at the computing hosts;

means for detecting over-utilization of one of the computing hosts and for issuing a warning when one of the computing hosts is over-utilized.

computing resources on a network including virtual machines for executing mobile agent software code;

means for transferring the mobile agents to the computing resources; and

means for automatically transferring execution code as well as stack trace and state information about each of the mobile agents responsive to detected over-utilization of one of the computing hosts from a central server to the computing resources, the computing resources receiving and executing one of the small tasks assigned to a mobile agent in the virtual machines using the execution code from the central server in response to a request from the available computing resources and the means for transferring execution code maintaining stack trace and state information about each of the mobile agents at a first computing host where none of the mobile agents are executing to allow each of the mobile agents to be reconstructed at a second computing host.

Claim 27 (cancelled)

28. (original) The ~~apparatus~~ computer system of claim 26 wherein the central server comprises a web server.

29. (original) The ~~apparatus~~ computer system of claim 26 further comprising means for monitoring execution of the small tasks.

30. (original) The ~~apparatus~~ computer system of claim 26 further comprising collaboration means for allowing the mobile agents to communicate and share information in real time.

31. (original) The ~~apparatus~~ computer system of claim 26, wherein the mobile agents execute in real time.

32. (original) The ~~apparatus~~ computer system of claim 26, further comprising storage means for storing real time state information about the mobile agents as the mobile agents execute at the computing resources.

33. (original) The ~~apparatus~~ computer system of claim 26 further comprising:
means for monitoring execution of the mobile agents at the computing hosts; and
means for detecting over-utilization of one of the computing hosts and for issuing a warning when one of the computing hosts is over-utilized.

34. (previously amended) A method for providing realistic thread migration which comprises:

- instantiating a mobile agent thread at a first computing host;
- processing said mobile agent thread at said first computing host;
- storing stack trace and state information about said mobile agent thread at a second computing host at which said mobile agent thread is not executing as said mobile agent thread executes at said first computing host;
- detecting an indication to migrate said mobile agent thread and in response to said indication;
- stopping execution of said mobile agent thread;
- automatically transferring the execution code for said mobile agent thread from a central server to a third computing host; and
- automatically transferring said stack trace and state information about said mobile agent thread to said third computing host.

Claims 35-38 (cancelled)

38. (previously amended) The method of claim 34 further comprising:

- receiving said stack trace and state information about said mobile agent thread at said third computing host;
- reconstructing said mobile agent thread at said third computing host using said stack trace and state information about said mobile agent thread; and

continuing processing of said mobile agent thread at said third computing host at a point at which execution of said thread was stopped at said first computing host.

39. (previously amended) The method of claim 38 wherein the step of automatically transferring said stack trace and state information about said mobile agent thread further comprises serializing the state information about said mobile agent thread.

Claims 40-41 (cancelled)

Claims 42-45 (cancelled)

46. (previously amended) The method of claim 1 further comprising reconstructing one or more of said mobile agents at said second computing host using said stack trace and state information stored at said first computing host.

47. (currently amended) The method of claim 11, wherein the step of transferring said one or more mobile agents to one or more available computing hosts comprises transferring data relating to said one or more mobile agents to said one or more ~~mobile~~ available computing hosts.

48. (previously amended) The ~~method~~ computer system of claim 26, wherein said means for transferring said mobile agents to said computing resources comprises means for transferring data relating to said mobile agents to said computing resources.

49. (previously amended) The ~~apparatus~~ computer system of claim 28, wherein the web server is located at a first organization.

50. (previously amended) The ~~apparatus~~ computer system of claim 49 further comprising an agent execution environment located at a second organization.

51. (previously amended) The ~~apparatus~~ computer system of claim 50 further comprising firewalls at said first and second organizations to allow execution code for said mobile agents to be exchanged between said first and second organizations.

52. (previously presented) The method of claim 34, wherein the steps are carried out without the need to explicitly write any additional software code to either initiate, manage, or facilitate the transfer of a mobile agent during said process of thread migration.